PIXINWAV: RESIDUAL STEGANOGRAPHY FOR HIDING PIXELS IN AUDIO

Margarita Geleta[†,§] *Cristina Punti*[⋆] *Kevin McGuinness*[⋆]
Jordi Pons[*] *Cristian Canton*[‡] *Xavier Giro-i-Nieto*[†,‡]

[†]Universitat Politècnica de Catalunya [⋆]Insight Centre for Data Analytics – Dublin City University
[*]Dolby Laboratories [§]University of California, Irvine
[‡]Institut de Robòtica i Informàtica Industrial (CSIC-UPC)

ABSTRACT

Steganography comprises the mechanics of hiding data in a host media that may be publicly available. While previous works focused on unimodal setups (e.g., hiding images in images, or hiding audio in audio), PixInWav targets the multimodal case of hiding images in audio. To this end, we propose a novel residual architecture operating on top of short-time discrete cosine transform (STDCT) audio spectrograms. Among our results, we find that the residual steganography setup we propose allows an encoding of the hidden image that is independent from the host audio without compromising quality. Accordingly, while previous works require both host and hidden signals to hide a signal, PixInWav can encode images offline—which can be later hidden, in a residual fashion, into any audio signal.

Index Terms— steganography, multimodal, deep learning

1. INTRODUCTION

Steganography (with Greek roots: "steganós" meaning covered, and "graphein" meaning writing) refers to the method of concealing a *container* signal embedding a *hidden* signal within a *host* signal. The resulting container signal may be sent through a publicly accessible channel in a way that the hidden signal stays inconspicuous to potential observers. Steganography has benefited from recent advances in deep learning, especially in the uni-modal front: image/video [1, 2] or audio [3]. In this work, we focus on the unexplored multi-modal case of hiding images in audio signals, such that the host signal is audio and the hidden signal is an image. Hiding images into audio allows the exploitation of existing audio infrastructures for image and video distribution. For instance, analog broadcast radio may transport the album cover of a song being played, loudspeakers in airports could distribute maps or visual messages for the hearing impaired, or video streams could be distributed to handheld devices of crowds located in areas with insufficient mobile network capacity. The proposed technical solution also has direct applications to watermarking, where the hidden information is related to its content, thus enabling potential provenance solutions; and to media forensics, where digital content must be analyzed to determine whether it is authentic, fake, or if it has been modified, so that backdoor attacks can be prevented [4].

Deep neural networks have attracted the attention of steganography researchers, as they have the potential to learn the best representations for hiding the secret signal within the host. Unlike classic steganographic methods that encode the hidden signal within the least significant bits of the host signal, deep learning approaches can compress and spread the secret signal's representation over all the available bits. Previous deep learning approaches [1, 2, 5, 6] have adopted architectures where both the host and hidden signals are fed into a deep neural network to construct the container signal. Our solution does not rely on this costly merge operation. Instead, our proposed PixInWav model relies on a simple (yet effective) residual architecture [7]: we train a neural encoder for the hidden image independently from the host, so that a fixed and learned representation can be simply added to the audio spectrogram to build the container signal—see Figure 1. Throughout our work, we show that it is not necessary to learn a different representation for each hidden signal depending on the host, since the same encoded representation of the secret image can be added to any audio.

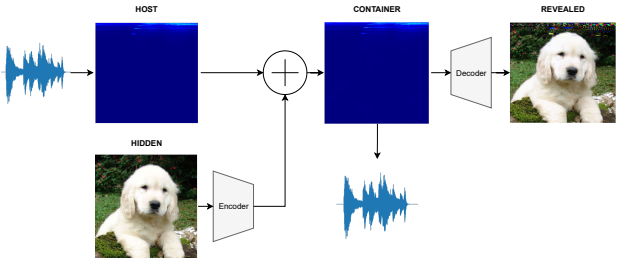


Fig. 1: PixInWav proposes learning independent image representations that can be hidden within audio signals.

Our contributions can be summarized as follows: (a) we explore, for the first time, a deep learning approach to hide images in audio signals; (b) we address this task via a novel residual architecture operating on top of a short-time discrete cosine transform (STDCT) audio representation; and (c) we show that encoded images can be created independently from the audio signal, such that they can be pre-computed per image and later added to any arbitrary audio signal. Finally, we also release code to reproduce our experiments:

https://github.com/margaritageleta/PixInWav

2. RELATED WORK

Hiding information in audio is a relatively unexplored research area. Early works on audio steganography relied on signal processing and audio coding [8, 9, 10, 11, 12], but recent advances rely on deep learning [3]. Many early methods encode the hidden signal within the perceptually least significant bits (LSB) of audio [9, 10, 11]. This strategy was adopted by the only precedent, up to the author's knowledge, of hiding images in audio [12]. However, they treat images as a generic digital signal. There was no learned visual representation, or any special choice because of the visual nature of the hidden message. Further, this work only provided very basic qualitative results, hiding a single image into a single audio clip in a completely *in silico* setup. Our experiments are much more extensive, exploring the effects of different types of noise.

To the best of our knowledge, no previous works used deep learning for hiding images into audio. Only Kreuk et al. [3] used neural networks for audio steganography, to send multiple speech recordings through a single host audio. Kreuk et al. [3] noted that steganographic vision-oriented models are less suitable for audio, and propose learning a steganographic function in the frequency domain. To this end, they employ the short-time Fourier transform (STFT) and describe the challenges associated with this complex transform (that is normally decomposed as magnitude and phase). Although many STFT-based audio models discard the phase, Kreuk et al. [3] argue that discarding the phase is not practical since the decoder will be forced to also infer the phase. To circumvent this challenge, Kreuk et al. [3] propose using differentiable STFT layers as part of their model. Our work also contributes to the discussion of which deep learning architectures are more suitable for audio steganography: (i) we propose using the short-time discrete cosine transform (STDCT, a real-transform), instead of the STFT (a complex-transform, with magnitude and phase) to avoid the above-mentioned challenges; and (ii) we propose a novel residual architecture that hides a secret image by adding a perceptually transparent perturbation to the host audio.

On the other hand, hiding information in image pixels has been extensively explored, with significant progress achieved by recent deep learning techniques [1, 2, 5, 13, 14, 15, 16, 17, 18]. Most deep image steganography techniques rely on convolutional neural network (CNN) encoders to hide the message within a host image, and a CNN decoder to recover the hidden message. Such systems are generally trained following a loss schema in which (i) the encoder is trained to minimize a distortion over the host image; and (ii) the decoder is trained to minimize the reconstruction loss over the (recovered) secret image. Hence, the output of the encoder (container) includes an image that is perceptually similar to the host image but contains a (hidden) secret image—that the decoder can recover.

3. METHODOLOGY

PixInWav follows the classic encoder-decoder paradigm composed of two networks, which are trained end-to-end, to hide images into STDCT spectrograms. The encoder hides an image into a host spectrogram in the shape of a (perceptually transparent) perturbation. The decoder is responsible for mapping the residually added perturbation back to an RGB image, and it is trained to minimize the reconstruction loss with respect to the revealed (or hidden) image. In our experiments, we apply different degrees of noise on the container audio, and assess its impact on the recovery of the secret image.

3.1. Residual Architecture

While previous deep steganography solutions have attempted to jointly learn a representation for both the host and hidden signals, we propose to learn a representation for the hidden image only, and then add this into the host audio spectrogram. This residual-based approach, inspired by the residual modules in ResNet [7], makes it straightforward for the optimizer to fit an approximate identity function of the host signal, since this signal does not need to undergo a series of transformations due to the steganographic embedding function, as has been the case in previous works [7]. Note that steganographic applications aim at learning an *almost identity* function of the host signal, such that the hidden signal is unnoticeable during transmission. Motivated by these ideas, we propose to simply add the encoded image to the host audio in a residual fashion.

Fig. 2 (d) shows the proposed PixInWav residual architecture, next to three other configurations we compared against in our ablation study in Section 4.2. PixInWav encodes the hidden image and adds it to the STDCT-spectrogram of the host audio. The resulting container (stego-audio) is the signal to be transmitted and whose distortion with respect to the host audio should be perceptually unnoticeable. At the receiver end, a decoder reconstructs the hidden image, ideally, with the minimum possible perceptual distortion. Both the encoder and decoders are 2D fully convolutional neural networks with skip connections, based on the U-Net [19]. The encoder part (hiding network) contains both a contracting part (downsampling step) and an expansive part (upsampling step). The contracting part is composed by two downsampling modules, each one consisting of two 3×3 convolutions with stride 2 and 4, respectively. Each of the convolutional layers is followed by a batch normalization and a Leaky ReLU activation function. The expansive part is composed by two upsampling modules, each one composed of two transposed convolutional layers and two convolutions with batch normalization. Each of these layers have a kernel size of 3×3 and include a Leaky ReLU activation function in between. The decoder (revealing network) is composed of the same number of convolutional layers.

The 3-channel RGB images are augmented to four channels by appending a zero channel, subject to a 2×2 pixel shuffle operation [20] to rearrange these four channels into the spatial dimensions. This operation distributes color information into the spatial domain, which makes it more straightforward for the encoding network to create residuals to be added to the spectrogram that maintain the relevant color information, and was shown necessary to obtain high-quality color reconstruction. At the output of the revealing network, the inverse pixel unshuffle operation is applied to rearrange the spatial information back into the color channels.

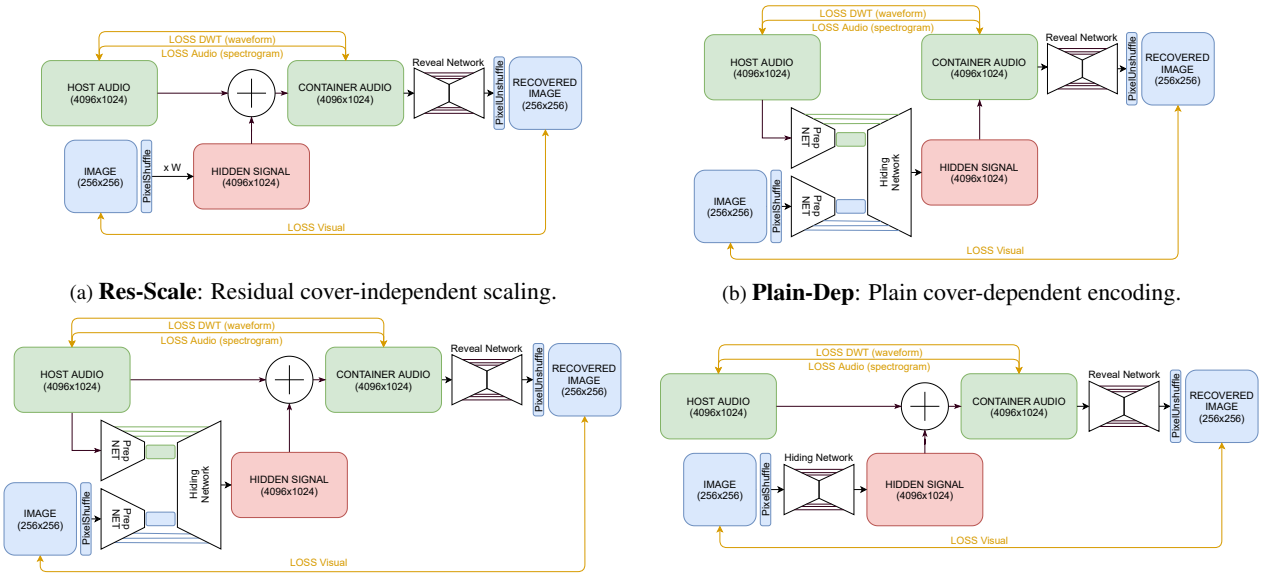
3.2. Audio Representation

Previous work on audio steganography relied on differentiable STFT layers to learn a steganographic function in the frequency domain [3]. In line with that, note that (STFT or STDCT) spectrograms are 2D audio representations that allow for a natural way to hide (in a residual fashion) images into audio—since one can exploit the 2D nature of spectrograms to hide images while preserving locality. For that reason, we discarded relying on a waveform-based model, because this would require encoding the image in a 1D signal. Further, note that the spectrogram representation we study (the STDCT) is a deterministic operation allowing perfect (inverse) reconstruction. Consequently, via employing spectrograms for residual audio steganography, we preserve the simplicity of not requiring an encoder for the host signal—but we expand our approach by allowing it to also encode locality.

As noted in previous section, using STFT-based setups can introduce difficulties related to phase reconstruction. To overcome this problem, we propose using the short-time discrete cosine transform (STDCT, a real-transform), instead of the STFT (a complex-transform, with magnitude and phase) as a simple but effective way to overcome phase-related issues. In short, the main difference between STFT and STDCT is the type of basis function used by each transform: the STFT uses a set of harmonically-related complex exponential basis, while the STDCT uses (real-valued) cosine basis [21]. Our setup relies on the type-2 DCT.

3.3. Loss function

PixInWav is trained with a loss function allowing a trade-off between: low distortion of the host audio, and the reconstruction quality of the hidden image. This can be expressed as a convex combination of two reconstruction losses with a trade-off hyperparameter $\beta \in [0, 1]$.



(a) **Res-Scale**: Residual cover-independent scaling.

(b) **Plain-Dep**: Plain cover-dependent encoding.

(c) **Res-Dep**: Residual cover-dependent encoding.

(d) **Res-Indep**: Residual cover-independent encoding (*PixInWav*).

Fig. 2: **Neural architectures for hiding images in audio spectrograms:** The proposed PixInWav architecture corresponds to the *Res-Indep* setup. In our ablation study, we compare against these three alternative solutions: *Res-Scale*, *Plain-Dep* and *Res-Dep*.

Let s be the hidden image, s' the revealed image, C the host spectrogram and C' the container spectrogram. The steganographic system is trained by minimizing the addition of the image and spectrogram reconstruction errors:

$$\mathcal{L}(s, s', C, C') = \beta \|s - s'\|_1 + (1 - \beta)\|C - C'\|_2. \quad (1)$$

The loss function adopts a simple mean squared error (MSE) for the reconstruction of the host audio, but uses the mean absolute error (MAE) to measure image reconstruction quality. Using both L2 in the image and audio domain delivered worse results.

In our experiments, we also added an additional term to equation (1), the soft dynamic time warping (DTW) discrepancy [22] (with $\gamma = 1$) between the host waveform and the container waveform. The term is modulated by a constant $\lambda = 10^{-4}$ to make it comparable in magnitude to the loss in (1), giving a total loss of:

$$\mathcal{L}_{\text{total}}(s, s', C, C') = \mathcal{L}(s, s', C, C') + \lambda \, \mathbf{dtw}_{\gamma}(c, c'), \quad (2)$$

where c and c' are the original and reconstructed waveforms (i.e., $c = \text{STDCT}^{-1}(C)$). This additional term encourages the temporal alignment between the host and container audios.

4. EXPERIMENTS

4.1. Setup

Dataset: The audio signals used in this study correspond to the FSDnoisy18K dataset [23]. This dataset contains 18,532 audio clips across 20 sound classes, depicting a large variety of sounds, such as voice, music, or noise. Since the duration of each clip is variable, we randomly select audios of approximately 1.5 seconds at 44,100Hz. We computed the STDCT transform with a frame length 2^{12} and a hop size of $2^6 - 2$. These hyperparameters where chosen to obtain a spectrogram with width and height being powers of 2, which allows for efficient computations. RGB images were sampled from the ImageNet (ILSVRC2012) dataset [24]. 10,000 randomly sampled images were used to train PixInWav, while validation results are reported over a non-overlapping partition of 900 images. Each RGB image was resized, cropped and normalized, resulting in a $256 \times 256 \times 3$ image, and paired with a randomly selected sound from the audio dataset.

Training details: The model was trained with Adam at a learning rate (lr) of 0.01 and a batch size of 1. Additional experiments with $lr = 0.1, 0.001$ did not converge. Leaky-ReLUs are set to $\alpha = 0.8$. The revealed image at the output is clipped in the range of $[0, 1]$ and denormalized back to the range of RGB values: $[0, 255]$.

Evaluation metrics: The inclusion of the image into the audio signal introduces a distortion, which we measure with the signal-to-noise-ratio (SNR) over the waveform—where the noise corresponds to the difference between the host and container audios. We adopted SNR as audio quality metric because it is widely used among related works [3, 8]. Analogously, the image also suffers a distortion as a result of encoding it into audio with the hiding network, and decoding it with the reveal network. We measure the visual distortion with the Structural Similarity Index (SSIM) [25], a metric that takes into consideration the perceptual properties of the human visual system. In addition, we also provide results in peak signal-to-noise-ratio (PSNR), to allow contrasting our values with the literature [5].

4.2. Results

Trade-off between audio-image distortions: PixInWav was trained with different distortion trade-offs between the host audio and the hidden image, governed by the β parameter of the loss function. The impact of β is studied from a quantitative (Fig. 3) and qualitative (Fig. 4) perspective.

As expected, lower β values preserve better the quality of the audio, while higher ones allow a better recovery of the hidden image, at the expense of a reduction of audio SNR. The obtained results demonstrate that PixInWav can meet perceptually acceptable quality

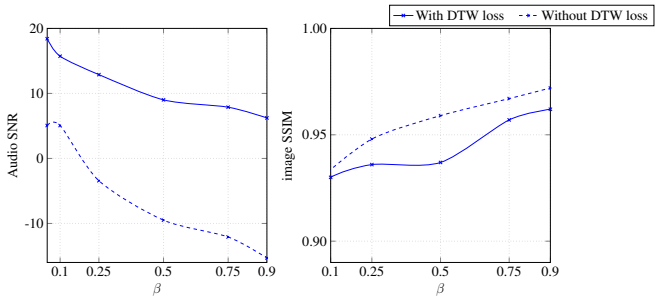


Fig. 3: Quality trade-off between the host audio (left) and the hidden image (right). x-axis: the hyperparameter β that controls the distortion between the two. Results after training for 8 epochs.

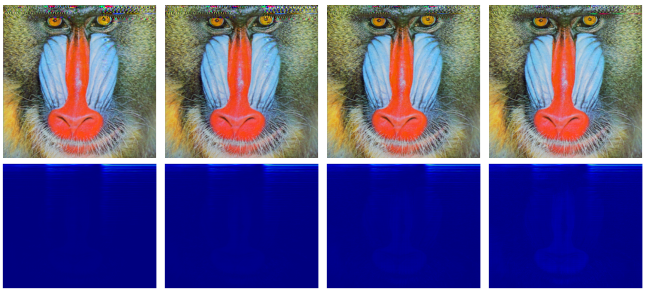


Fig. 4: Effect of β on the image and spectrogram. Each column refers to a $\beta = \{0.05, 0.1, 0.5, 0.9\}$, respectively.

standards for both the host audio and the hidden image. As a reference, it is considered that listeners will barely notice any distortion when the audio SNR is above 20 dB, and intelligibility will still be reasonable at 0 dB SNR (speech energy and noise energy being the same) [26]. In the remainder of our experiments, we set a β of 0.05, which corresponds to an average audio SNR of 18.26 dB and an average SSIM of 0.921 for a model trained during 8 epochs.

Dynamic time warping loss on audio: Fig. 3 plots the corresponding SNR and SSIM values if the DTW loss term was not included. The results show the importance of this term, as removing it drops the audio SNR more than 10 dB, moving below the 0 dB case for most tested β. On the other hand, the DTW loss term applied over the audio signal actually introduces only a small distortion over the images, unnoticeable for a human.

Ablation study: The *Res-Indep* architecture we proposed for PixInWav (in Section 3.1) is now compared with the other three approaches we depict in Fig. 2: *Res-Scale*, *Plain-Dep* and *Res-Dep*. *Plain-Dep* is used to compare our residual approach with a classic feedforward encoder-decoder architecture. *Res-Dep* shows the effect of conditioning the hiding network on both the image and the host audio. And *Res-Scale* is included to check if the hiding network is not simply uniformly encoding the image signal in the low-order bits of the host signal. We compared the four solutions based on quantitative (see Table 1) and qualitative results (in Fig. 5).

The best quantitative and qualitative results are obtained with the *Res-Dep* and *Res-Indep* solutions, both achieving similar performance and learning behaviour. Nevertheless, *Res-Indep* is a much more scalable solution because it uses an image representation that does not depend on the host audio, while *Res-Dep* requires the computation of a specific transformation for each audio snippet. Regarding the other two options, *Res-Scale* prevents the image from being recovered, while *Plain-Dep* fails at transmitting the audio.

Architecture	Audio SNR ↑	Image SSIM ↑	Image PSNR ↑
Res-Scale	14.66	0.5414	19.52
Plain-Dep	−0.73	0.971	32.70
Res-Dep	18.80	0.919	27.29
Res-Indep	18.33	0.923	27.37

Table 1: **Ablation study:** Audio and image quality metrics for a fixed $\beta = 0.05$ with *PixInWav* and the three considered baselines. Results after training for 5 epochs.

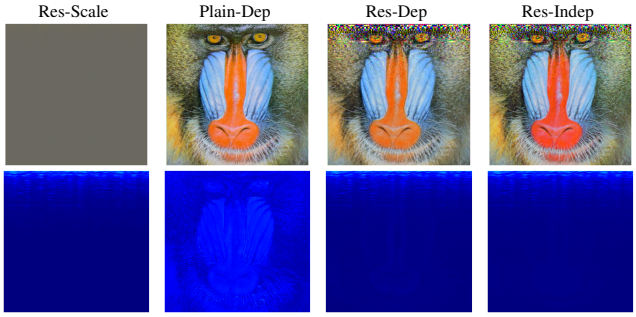


Fig. 5: **Qualitative comparison between architectures:** *Res-Scale* fails in transmitting the image and *Plain-Dep* allows a visible interference of the image over the spectrogram. On the other hand, *Res-Dep* and *Res-Indep* provide similar qualitative results, while the later reproduces more similar colors to the original image.

Embedding capacity: We transmit a 256×256 color image (3 channels) of 8 bits per pixel. Each audio clip contains 67,522 samples at a sampling rate of 44,100 Hz, which corresponds to 1.53 seconds per clip. These values result in a transmission rate of 988 Kbps.

Computational cost of the method: Both the hidden and reveal networks are identical and contain 482,090 parameters. The total computation cost of encoding, adding and decoding an image is of 197.31 GMAC (Giga multiply-accumulate operations).

5. CONCLUSIONS

This paper presents pioneering work on deep multimodal steganography in which we have explored the transmission of visual information over audio. Our residual approach to deep steganography proposes obtaining an encoding of the hidden image that can be directly added to the host audio. Importantly, those hidden image encodings can be computed independently from the host audio, which makes the system much more scalable than previous approaches. We also found that the following strategies were beneficial for training: (i) adding a DTW loss term, and (ii) employing the pixel shuffle layer for encoding the hidden image.

6. ACKNOWLEDGEMENTS

Work partially supported by the European Union through the Erasmus+ student mobility program, Science Foundation Ireland (SFI) under grant numbers SFI/15/SIRG/3283 and SFI/12/RC/2289_P2, and the Spanish Research Agency (AEI) under project PID2020-117142GB-I00 of the call MCIN/ AEI /10.13039/501100011033.

7. REFERENCES

[1] Matthew Tancik, Ben Mildenhall, and Ren Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[2] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.

[3] Felix Kreuk, Yossi Adi, Bhiksha Raj, Rita Singh, and Joseph Keshet, "Hide and speak: Towards deep neural networks for speech steganography," *Proc. Interspeech 2020*, pp. 4656–4660, 2020.

[4] Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, and Xinpeng Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[5] Shumeet Baluja, "Hiding images within images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1685–1697, 2019.

[6] Xintao Duan, Kai Jia, Baoxia Li, Daidou Guo, En Zhang, and Chuan Qin, "Reversible image steganography scheme based on a U-Net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] Rully Adrian Santosa and Paul Bao, "Audio-to-image wavelet transform based audio steganography," in *47th International Symposium ELMAR, 2005.* IEEE, 2005, pp. 209–212.

[9] Mohammed Salem Atoum, Subariah Ibrahimn, Ghazali Sulong, Akram Zeki, and Adamu Abubakar, "Exploring the challenges of mp3 audio steganography," in *2013 International Conference on Advanced Computer Science Applications and Technologies*. IEEE, 2013, pp. 156–161.

[10] N. Cvejic, "Algorithms for audio watermarking and steganography," 2004, Department of Electrical and Information Engineering, Information Processing Laboratory, University of Oulu.

[11] Kadir Tekeli and Rifat Asliyan, "A comparison of echo hiding methods," *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, vol. 1, pp. 397–403, 2017.

[12] Dalal N Hmood, Khamael A Khudhiar, and Mohammad S Altaei, "A new steganographic method for embedded image in audio file," *International Journal of Computer Science and Security (IJCSS)*, vol. 6, no. 2, pp. 135–141, 2012.

[13] Linjie Guo, Jiangqun Ni, and Yun Qing Shi, "An efficient jpeg steganographic scheme using uniform embedding," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2012, pp. 169–174.

[14] Vojtěch Holub and Jessica Fridrich, "Designing steganographic distortion using directional filters," in *2012 IEEE International workshop on information forensics and security (WIFS)*. IEEE, 2012, pp. 234–239.

[15] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1, 2014.

[16] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *International Workshop on Information Hiding*. Springer, 2010, pp. 161–177.

[17] Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont, "Steganography using a 3 player game," *CoRR*, vol. abs/1907.06956, 2019.

[18] Jarno Mielikainen, "LSB matching revisited," *IEEE signal processing letters*, vol. 13, no. 5, pp. 285–287, 2006.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[20] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[21] J. Casebeer S. Venkataramani and P. Smaragdis, "End-to-end source separation with adaptive front-ends," 2017.

[22] Marco Cuturi and Mathieu Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds., International Convention Centre, Sydney, Australia, 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 894–903, PMLR.

[23] Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra, "Learning sound event classifiers from web audio with noisy labels," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 21–25.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "ImageNet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[26] Dan Ellis, "Berkeley international computer science institute (icsi) speech faq," 2009.